ANL-ARC-299

SAS4A/SASSYS-1 Modernization and Extension

Nuclear Engineering Division

About Argonne National Laboratory
Argonne is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC under contract DE-AC02-06CH11357. The Laboratory's main facility is outside Chicago, at 9700 South Cass Avenue, Argonne, Illinois 60439. For information about Argonne and its pioneering science and technology programs, see www.anl.gov.

ANL-ARC-299

SAS4A/SASSYS-1 Modernization and Extension

prepared by

T. H. Fanning, A. Brunett, T. Sumner, and N. Stauff
Nuclear Engineering Division, Argonne National Laboratory

September 26, 2014

ABSTRACT

SAS4A/SASSYS-1 is a simulation tool used to perform deterministic analysis of anticipated events as well as design basis and beyond design basis accidents for non light-water reactors. This report summarizes ongoing tasks to modernize the SAS4A/SASSYS-1 code system to improve internal data management and to update several other code modules. The motivation for performing these updates stems from the relevance of SAS4A/SASSYS-1 to a number of U.S. Department of Energy programs as well as domestic and international collaborations.

In addition to code modernization, important new extensions have been introduced to address needed capabilities. The most significant of these is the development of a coupling interface that allows SAS4A/SASSYS-1 to interact with external power conversion system models. This capability will be used to couple SAS4A/SASSYS-1 with the CO_2 Brayton cycle Plant Dynamics Code, which will give analysts the ability to assess full plant responses to a variety of load demands or system upset conditions.

Other modeling extensions include updates to the control system module to provide greater development flexibility and to provide users with the capability to plot signal data that are produced during a simulation. Reactivity feedback coefficients are now easier to import into SAS4A/SASSYS-1 as a result of a collaboration with the NEAMS development team that lead to improvements in the PERSENT perturbation theory transport code. Spatial kinetics coupling can better handle some of the more detailed cross section data generated by the MC^2-3 code, however some issues still remain for handling delayed neutron precursor data.

SAS4A/SASSYS-1 has a slowly growing user base that will strengthen the promotion of advanced reactor concepts such as sodium cooled fast reactors. These external collaborations have also produced improvements in the SAS4A/SASSYS-1 code that have provided real benefit for DOE users, particularly for the ASTRID collaboration. Additional users will help solidify DOE's leadership role in fast reactor safety both domestically and in international collaborations.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. Introduction

SAS4A/SASSYS-1 is a simulation tool used to perform deterministic analysis of anticipated events as well as design basis and beyond design basis accidents for advanced liquid-metal-cooled nuclear reactors. With its origin as SAS1A in the late 1960s, the SAS series of codes has been under continuous use and development for over forty-five years and represents a critical investment in safety analysis capabilities for the U.S. Department of Energy. This report documents fiscal year 2014 activities that were carried out to modernize data management and code structure, to include coupling with external balance of plant models, and to improve control system, reactivity feedback, and spatial kinetics modeling. This report satisfies the deliverable for the Level 3 milestone M3AR-14AN1701042, "Progress of SAS4A/SASSYS-1 Modernization."

The rapid growth in the availability of modern, object-oriented programming practices greatly facilitates code development and maintenance. Although much of the SAS4A/SASSYS-1 source code reflects traditional software development practices, modern code development practices are being used to restructure critical parts of data management and to implement extended capabilities that address programmatic needs.

Modernization and extension of the SAS4A/SASSYS-1 code system are motivated by the relevance of its simulation capability to a number of U.S. Department of Energy programs as well as domestic and international collaborations.[1] Active programs and collaborations that currently use SAS4A/SASSYS-1 include the following:

- EBR-II IAEA Benchmark: The DOE-NE Advanced Reactor Concepts program is supporting a high-profile Coordinated Research Project with the International Atomic Energy Agency based on the Shutdown Heat Removal Tests conducted at EBR-II. Both protected (SHRT-17) and unprotected (SHRT-45R) loss-of-flow tests are part of the benchmark activity. SAS4A/SASSYS-1 models of both tests are being developed to provide results under the CRP.[2,3]
- ASTRID Collaboration with CEA: An implementation agreement has been established between the U.S. DOE and the Commissariat à l'énergie atomique et aux énergies alternatives (CEA) of France for cooperation in low carbon energy technologies. One purpose of the agreement is to evaluate the safety performance of the ASTRID reactor design. DOE performs these evaluations using the SAS4A/SASSYS-1 safety analysis code.[4]
- CIAE Bilateral Collaboration: The DOE-NE Office of International Nuclear Energy Policy and Cooperation has established the U.S.–China Bilateral Civil Nuclear Energy Cooperative Action Plan with the China Institute of Atomic Energy. Joint activities under the action plan include model development and safety analyses of the China Experimental Fast Reactor using SAS4A/SASSYS-1.[5] CIAE expects to perform whole-plant transient testing in the fall of 2014.
- AdvSMR Passive System Reliability: The DOE-NE Advanced Reactor Concepts program has conducted an evaluation of risk assessment methods that could be applied to passive safety systems.[6] One of the recommendations is to perform

dynamic probabilistic risk assessment methods. Plans include coupling the ADAPT tool[7] to SAS4A/SASSYS-1 to perform multiple dynamic event-tree simulations for SFR passive systems.

- TerraPower TWR Reactor Concept: TerraPower, LLC has licensed the SAS4A/SASSYS-1 source code to perform safety analysis studies for their "Traveling Wave Reactor" concept. TerraPower also funds code development activities that improve the modeling capabilities of SAS4A/SASSYS-1 (see Section 5).
- KAERI PGSFR: The Korean Atomic Energy Research Institute acquired a license for SAS4A/SASSYS-1 to perform safety analysis and model development for the "Prototype Generation-IV Sodium Fast Reactor". KAERI is supporting metallic fuel performance and severe accident model developments that will be incorporated into SAS4A/SASSYS-1.
- KINS: The Korea Institute of Nuclear Safety is an independent regulatory expert organization that supports the Nuclear Safety and Security Commission (NSSC) in Korea. KINS holds a license for SAS4A/SASSYS-1 to support the regulatory obligations over the PGSFR project (above).
- KTH ELECTRA LFR Concept: The Royal Institute of Technology (Kungliga Tekniska Högskolan) in Stockholm Sweden holds an academic license for SAS4A/SASSYS-1 to perform natural circulation design performance studies of their ELECTRA lead-cooled fast reactor concept.

Additional license requests have been received during 2014 and include the following:

- Korea University requested a license for SAS4A/SASSYS-1 following a seminar provided by Adrian Tentner during a visit to Seoul. Following execution of a license, version 5.0.3 was distributed in late August.
- University of Illinois at Urbana-Champaign: Professor Kozlowski has requested an academic license for SAS4A/SASSYS-1 with the intent of performing sensitivity analysis and uncertainty quantification using tools such as DAKOTA.
- University of Michigan: Professor Downar has requested an academic license with the intent of using SAS4A/SASSYS-1 in senior design projects to evaluate thermal hydraulic performance of SFR designs.
- LeadCold Reactors: Professor Wallenius (KTH, see above) has formed a startup company with the vision of deploying natural circulation lead-cooled reactors in arctic communities in Canada. LeadCold is seeking a commercial license for SAS4A/SASSYS-1 to evaluate their SEALER-3 (Swedish Advanced Lead Reactor) concept.

In the sections that follow, the modernization and extension work that was carried out during FY14 is described in detail. A brief summary of additional work supported by third-party users is also provided to give a broader perspective on overall SAS4A/SASSYS-1 developments. For more background on SAS4A/SASSYS-1 and details on previous modernization activities, see the FY13 report.[8]

2. Modernization and Extensions

Three key areas were the focus of FY13 modernization activities. These include 1) continued modernization of the data management strategy to facilitate ongoing and future code development activities, 2) development of a capability to couple SAS4A/SASSYS-1 to external power conversion system models, such as available models for a CO_2 Brayton cycle, and 3) improvements to key modules, such as the control system and reactivity feedback models, to provide capabilities needed in other DOE programs as described above.

2.1 Data Management Improvements

SAS4A/SASSYS-1 was originally developed for computing architectures with extremely limited memory capacities compared to current hardware. Early code development practices revolved around minimizing memory usage. A fundamental strategy was to overlay, or reuse, the same memory locations for multiple different models. This strategy made it difficult to add new models or extend existing models without risking corruption to memory locations used by other models.

Even though this strategy came with serious drawbacks, it was essential for small memory machines. Consumer computing hardware today has three orders of magnitude more memory than in the 1990s, so now these drawbacks impose limitations on code development and performance with no return benefit. These limitations include

- Maintainability: Code improvements that change the length or interpretation of the shared memory location can have unintended, far-reaching consequences. Any data following the change can shift to different memory locations. This affects all models, even unrelated ones, that expect the shared memory to have a fixed ordering.
- Scale: In theory, large-scale problems could be simulated with SAS4A/SASSYS-1. However the current data management strategy is limited to 32-bit addressing, which imposes a practical upper limit of around 10,000 channels. This means full-core sub-channel analysis is not yet possible.
- Speed: Even though overlaying data in memory is much faster than data transfer to disk, it is still slower than direct access. This is especially a problem when severe accident conditions are modeled, because small time steps require more frequent data transfers that severely impairs code performance.
- Parallelism: Because channels are generally independent, they could be solved in parallel. The current channel model depends on a single, shared memory location, so simultaneous use is not possible, and multi-threaded parallelism is not practical.

To remove these limitations, a number of significant changes are being implemented in the data management strategy in SAS4A/SASSYS-1. Updates that improve code maintainability are being prioritized ahead of scale, speed, and parallelism. To this end, updates have focused on eliminating data overlays and developing object-oriented data modules that encapsulate the data used by different models into concrete user-defined data types with well-defined structures. New or existing features of SAS4A/SASSYS-1 that use the new data modules can alter them without concern for disrupting other features of the code because each data module is independent of the others.

This seemingly simple approach required significant effort to develop, implement, and verify. As described above, changes to the shared memory structure can disrupt the entire code. Relocating one variable at a time was rejected as being too time consuming since there are thousands of variables that can occupy the shared memory at any given time. Instead, blocks of data were moved. In FY13, all the data blocks associated with model definitions were transferred to object oriented models. The first major block associated with simulation results (known as COMC) was also updated, however a problem associated with sub-channel simulations was left unresolved.

The issue with COMC caused the sub-channel model for the Advanced Burner Test Reactor to hang only three seconds into a transient simulation. In FY14, this issue was finally identified and resolved. Because the sub-channel solver is manipulating data for several hundred coolant channels simultaneously, it was particularly prone to data corruption as the new data models were implemented. In this case, the solver was attempting to reduce time step sizes in order to maintain accuracy, but the time step cutback was being set on the wrong channel. As a result, the solution would never converge and the code would hang.

In addition to COMC, the two data packs defined for severe accidents, COLC and PLUC, have also been updated. COLC contains additional thermal-hydraulic data primarily needed for the sodium boiling models, and PLUC contains all of the channel-dependent data used for in-pin fuel melting and relocation and ex-pin fuel expulsion.

COMC and COLC have been combined into a new module called “SinglePin”, as the data represents all the thermal-hydraulic simulation data defined for a single channel in SAS4A/SASSYS-1. Data from PLUC is now contained in a module called “SPMotion”, which represents simulation data for fuel melting and relocation within a single channel. These data types are now dynamically allocated according to the number of channels in a problem and whether the severe accident models are to be invoked. All data modules also manage their own *restart* state: periodically throughout a simulation, each data module will write its contents to disk so that a subsequent simulation can be started from those conditions.

The only channel-dependent data block that still uses the old data management strategy is DEFC/DEFORM5, which contains transient fuel performance data. Current plans are to update this data block in FY15. The challenge with this data is that it uses a *dual* overlay. Like the other channel-dependent blocks, the fuel performance data uses an overlay on a per-channel basis. However the same memory locations can also be interpreted differently depending on whether the channel contains oxide fuel (DEFC) or metallic fuel (DEFORM5). Separating this second overlay will be somewhat like separating salt and sugar. Fortunately, modern object-oriented programming concepts can handle this situation with *polymorphic* variables. A polymorphic variable is one whose data type may vary at runtime. Support for this feature was introduced in the Fortran 2003 standard and extended in the Fortran 2008 standard, but it has taken compiler vendors some time to fully implement these capabilities.

SAS4A/SASSYS-1 also uses many *channel-independent* data blocks. Although they do not overlay shared memory locations, they are still implemented based on outdated and error-prone practices. The blocks INEUTR, RNEUTR, and ANEUTR, which contain all spatial-kinetics input data and material compositions, are now implemented as new data modules.

Use of the new modules revealed a number of missing variable declarations that could not be detected before. The block PKFLT, which contains point kinetic and decay heat precursor information, has been converted to a module that is dynamically allocated. This provides an opportunity to remove the restriction on the number of delayed neutron precursor families, which is currently fixed at six.

Finally, an important change related to data integrity is the implementation of "IMPLICIT NONE" throughout the code. This change eliminates implied variable declarations and required the addition of explicit definitions for all data. As a result, additional compiler diagnostics can now be used to detect common errors such as the use of uninitialized variables.

2.2 Coupling with Advanced Power Conversion Systems

As reactor designs have improved in efficiency and performance, the power conversion cycle (the last stage in determining plant efficiency) has also undergone numerous innovations that improve cycle efficiency, safety and cost. These advanced power conversion systems (PCS) are typically paired with higher-efficiency reactor designs, such as high temperature gas or liquid metal-cooled reactors. While SAS4A/SASSYS-1 does maintain robust whole-plant modeling capabilities, its balance of plant model is restricted to a conventional Rankine cycle. Since the system code has long been the industry standard for liquid metal-cooled reactor safety analyses, it is a natural next step for the software to include advanced PCS modeling capabilities.

In previous work, SAS4A/SASSYS-1 had been "coupled" to a super-critical CO_2 PCS code through the time-consuming process of code restarts at every time step.[9] While useful in assessing the feedback effects between the reactor and this specific process plant, the inefficient means of code linking is not suitable as a general interface between SAS4A/SASSYS-1 and a PCS code. In FY14, the ability to interface with any PCS code has been implemented directly in SAS4A/SASSYS-1, thereby eliminating the need for restarts. Incorporation of this interface enables the system code to be extremely flexible in that the effects of any advanced power conversion cycle on a primary system can now be analyzed.

2.2.1 Implementation

To couple SAS4A/SASSYSA-1 with an external PCS code, a new module (ExtHX – <u>Ext</u>ernal <u>H</u>eat E<u>X</u>changer) was created within PRIMAR-4. The ExtHX module interfaces with a PCS code through input/output files. This level of encapsulation prevents data corruption within SAS4A/SASSYS-1 by isolating data packs from external software and limits the size of data transfer by providing only the necessary information to the PCS code. Use of file-based I/O is not the most efficient means of data transfer, but the module has been written such that only two simple subroutines (related to reading, writing data) need to be modified to support other protocols, such as via network or socket communications.

For simplicity and robustness, the coupling points for this module are in parallel to the existing steam generator models, since an external PCS code will couple through a heat exchanger to the rest of the plant. To that end, a new element type was created (ITYPEL(IELL) = 14, external heat exchanger) so that progression through the PRIMAR-4 subroutines could mimic that of a steam generator, but diverge and follow ExtHX subroutines when necessary.

2.2.2 Module Progression

At a high level, the ExtHX operates on the following cycle, shown in Figure 1:

- Fluid temperature and mass flow rate for each coupled flow element (ExtHX) are determined by SAS4A/SASSYS-1.
    - This data is written to a file which is then read by the PCS code.
    - SASSA/SASSYS-1 waits for new data from the PCS code.
    - Once data is available, it is read into memory and made available to PRIMAR-4 subroutines.
- SAS4A/SASSYS-1 completes remainder of time step calculations.

Prior to any transient time steps, the ExtHX module is initialized with all other PRIMAR-4 variables in the SSPRM4 subroutine. The initialization function allocates storage arrays for the relevant variables, opens log and data transfer files, and checks the status of the PCS code. If no errors are returned, SAS4A/SASSYS-1 execution will continue with steady state initialization, where ExtHX steady state calculations will be called as necessary.

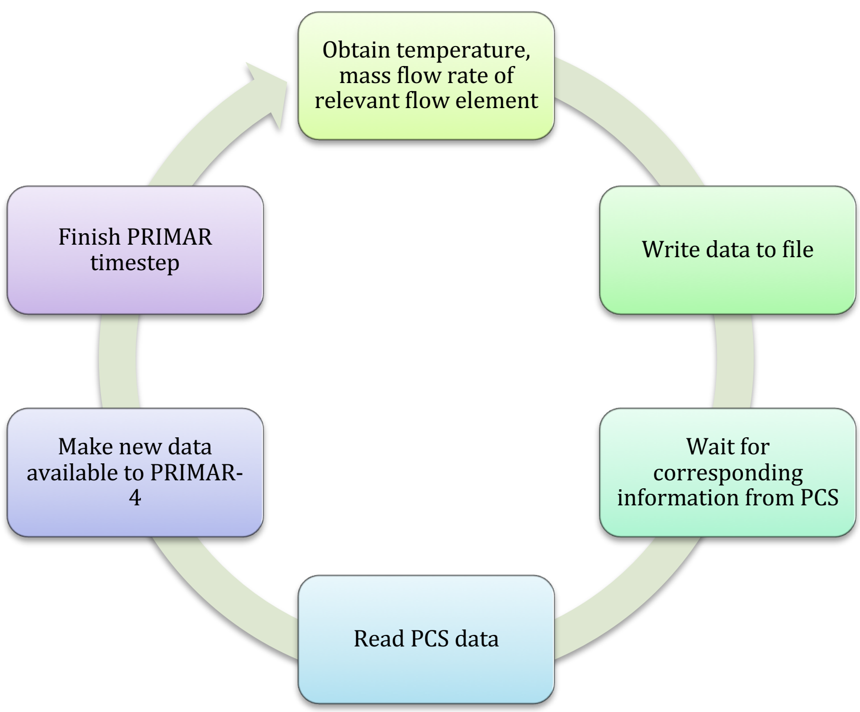

Figure 1: High level flow schematic of ExtHX processes. This cycle repeats at each PRIMAR subinterval.

For a steady state time step, a predetermined thermal loss that is the balance of all other heat losses and sources in the model is transferred to the PCS code to maintain the total energy of the system at steady state. For a null transient that treats component to component heat transfer, the same thermal loss is reported with a status message indicating a null time step. It is expected that the user will implement alternative PCS calculations if invoking a SAS4A/SASSYS-1 null transient.

Once all steady state variables have been computed, SAS4A/SASSYS-1 will then continue onto transient calculations. STEPTM, the subroutine responsible for subinterval transient temperature calculations in PRIMAR-4, will trigger a PCS calculation for any ExtHX flow element. During a transient, temperature, mass flow rate, timing of the PRIMAR-4 subinterval (for maintaining time step synchronization between codes) and a status message are written to a data file. After writing this data, SAS4A/SASSYS-1 waits for an analogous file containing the flow element outlet temperature, gravity head, and status message to be created by the PCS code. SAS4A/SASSYS-1 reads the file and makes the updated information available to other PRIMAR-4 subroutines. From this point, the transient time step will continue as before and the ExtHX module will not be called again until the code requests calculation of the state of another ExtHX flow element. The framework for use of the ExtHX module during a SAS4A/SASSYSA-1 restart has been inserted into the system code, but has not yet been tested. It is expected that during a restart calculation, the ExtHX module will be initialized and perform transient time steps as if it were a normal calculation (i.e. the subroutines utilized in a normal and restart calculation are the same).

2.2.3 Input Requirements

To utilize the ExtHX capability, four types of input (see Table 1) are required for the SAS4A/SASSYS-1 input file in PRIMAR-4 integer input (Block 3, INPMR4). The total number of input fields is dependent on the number of heat exchangers or interface points the user desires (i.e. the ExtHX module is capable of modeling up to eight externally-treated heat exchangers). The required input types are: *nEHX,* the number of interface elements between SAS4A/SASSYS-1 and the PCS code; *EHXelem(iEHX)*, the element number of each interface point; *EHXseg(iEHX)*, the segment number associated with each interface point; and *EHXtgp(iEHX)*, the temperature group number associated with each interface point. While segment numbers and temperature groups could have been obtained internally from other SAS4A/SASSYS-1 user input, the last two input types were included so that simple input checking on the ExtHX user input could be performed. Locations within INPMR4 are shown in Table 1.

Table 1: SAS4A/SASSYS-1 input required for ExtHX module.

Location	Symbol	Description
1413	nEHX	Total number of interface elements.
1414	EHXelem(iEHX)	Element number of each interface point.
1422	EHXseg(iEHX)	Segment number associated with each element.
1430	EHXtgp(iEHX)	Temperature group number associated with each element.

2.2.4 Demonstration Cases

Implementation of this module is shown for two test cases for the AFR-100, an advanced small modular reactor design. In these cases, the reactor is not paired with any specific PCS, so it should be emphasized that this is solely a functionality demonstration, and not an analysis of the AFR-100 design or the performance of a PCS. In both demonstration cases, the reactor remains at full power for the entire simulation.

In Case 1, discrete changes in the heat removal capacity of the PCS are implemented over several hours to emulate intentional programmatic changes in the PCS. It is assumed that the PCS trips at 100 s and heat removal is reduced by 50%. The system remains at 50% for 1 hour, then an additional 50% of capacity is lost for 30 min. Heat rejection is restored to 50% capacity for 30 min, then to 75% capacity for 1 hour. At 10800 s into the PCS transient, capacity of the PCS is restored to 100%. Results for the effect of discrete PCS capacity changes on interface element inlet and outlet temperatures and core outlet temperature are shown in Figure 2.

In Case 2, a cosine wave is imposed on the PCS heat rejection capacity to emulate dynamic loading on the primary system. In this case, the normalized heat rejection is assumed to cycle between 100% and 50% over a period of 1 hour. The effect of continuous changes in PCS capacity on the inlet and outlet temperatures of the PCS interface element and the propagation to the primary system is shown in Figure 3.

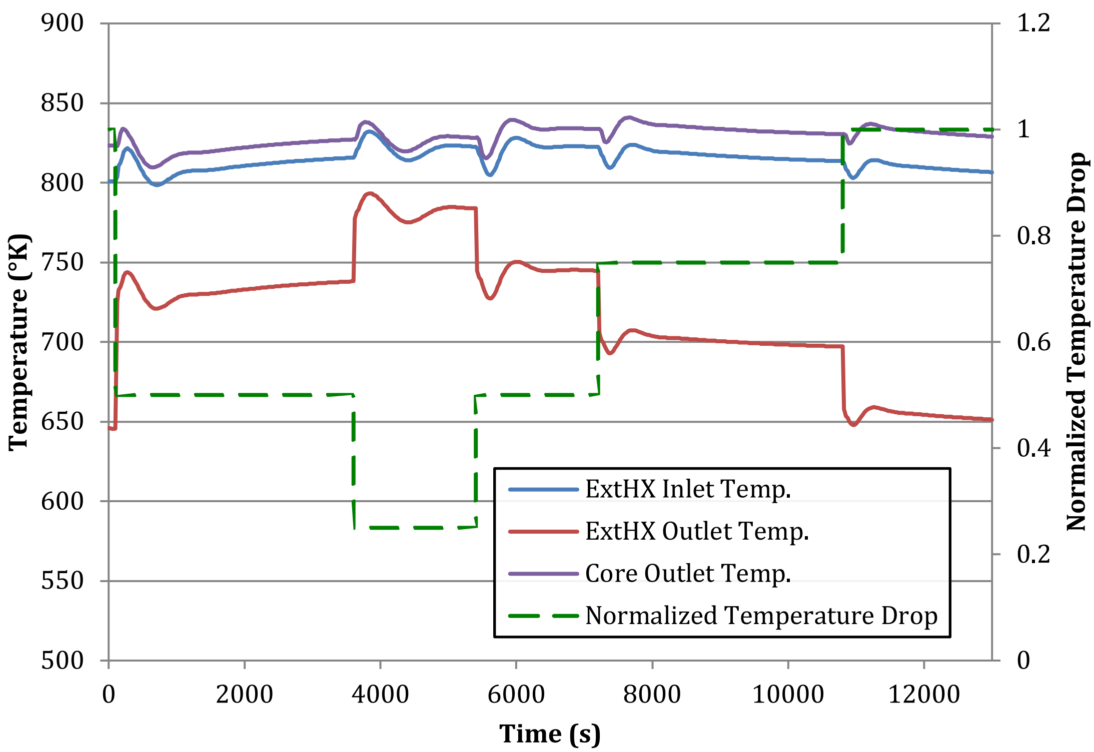

Figure 2: Inlet/outlet temperatures of PCS interface point and core outlet temperature for discrete changes in PCS capacity (Case 1).

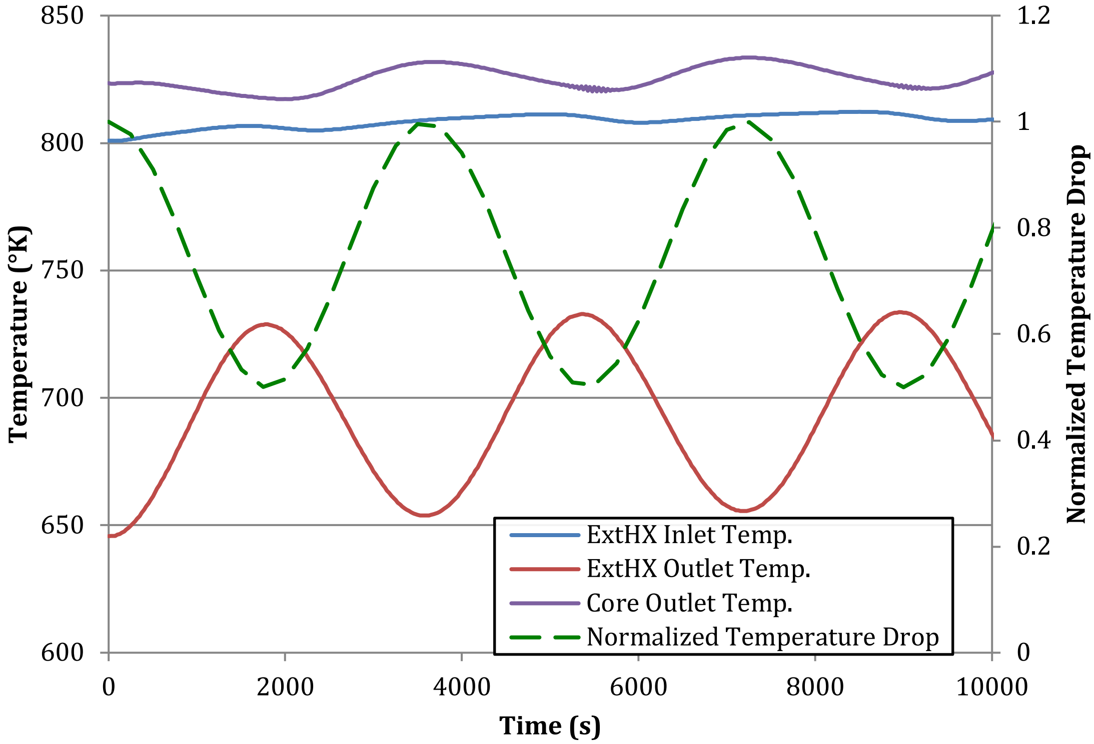

Figure 3: Inlet/outlet temperatures of PCS interface point and core outlet temperature for cosine imposition on PCS capacity (Case 2).

This work will continue into the next fiscal year in order to prepare the coupling between SAS4A/SASSYS-1 and the CO_2 Brayton cycle code PDC. To do this, however, some platform-specific dependencies need to be eliminated in the PDC code.

2.3 Other Modeling Extensions

A number of additional, smaller-scale modeling developments have been implemented in SAS4A/SASSYS-1 to support simulation needs for other DOE-NE programs. These improvements are a direct result of user feedback. The developments summarized below include 1) improvements to the control system model, 2) changes to the way reactivity feedback coefficients are prepared, 3) development of a preprocessor for spatial kinetics input, and 4) fixes for steady-state initialization.

2.3.1 Control System Updates

The control system provides users greater flexibility to model the actions and responses of various components during transient simulations. Networks of mathematical blocks can be created to control plant variables in the simulation, such as valve loss coefficients or externally driven reactivity feedback effects. Currently, the control system is being utilized for two important international collaborations: the ASTRID collaboration with CEA and the PGSFR collaboration with KAERI.

For the ASTRID collaboration, the control system is used to improve the standard control rod driveline expansion reactivity feedback model by accounting for the expansion of additional components not treated by the default model. For the PGSFR collaboration, the control system is used to model the response of the reactor protection system when plant safety limits are exceeded. In both cases, users need to be able to plot transient signal data from the control systems, and that capability was not available.

Updates to the control system data management address how the data is stored and how other parts of the code access that data. A new object-oriented data module, CNTL_Data, now acts as a controller for the control system. Data for the control system has been moved out of global common blocks and into a dynamically allocated data type in the module. The module also handles the generation of a binary data file that contains transient signal data for post-processing and plotting.

Access to data is now limited to procedures within the control system. Other parts of SAS4A/SASSYS-1 call type-bound procedures that act as wrapper functions. There are several benefits to using wrapper functions. First, the control system is isolated from the rest of the code, increasing code modularity. Other models in SAS4A/SASSYS-1, such as PRIMAR-4 or DEFORM, do not have direct access to the control system data. Changes can be freely made to the control system without disrupting other models and vice versa. This provides greater flexibility for future code enhancements and improves code maintainability.

Second, by creating type bound procedures that act on an individual instance of the control system, multiple instances of the control system could be created. Currently, only one instance is supported, but the use of multiple controllers would allow users flexibility to define more detailed and complex systems. For example, one controller could be defined to represent reactivity feedback behavior and a second controller could be defined to represent the reactor protection system. The models for each controller could be expanded and modified without introducing conflicting signal numbering schemes between the two models. This could improve collaboration by allowing multiple users to create control system models independently from the others.

2.3.2 Reactivity Feedback Coefficients

Neutronic feedback coefficients such as fuel, structure, and coolant density effects are traditionally estimated as changes in reactivity in units of $\Delta k/k$. Results are always relative to some material composition change. In SAS4A/SASSYS-1, reactivity coefficients must be normalized with respect to the mass of the composition change and have units of $(\Delta k/k)/kg$. Traditionally, reactivity feedback coefficients were produced by VARI-3D and had to be renormalized by hand, which was a time-consuming (and error-prone) process requiring the safety modeler to calculate the distribution of mass perturbations based on composition changes used by the analyst.

PERSENT is a new perturbation theory code based on the neutron transport equation. In collaboration with the NEAMS developers, PERSENT has been updated to determine the mass changes directly from the perturbed compositions and to renormalize the reactivity effects on a per unit mass basis. These coefficients may then be entered directly into SAS4A/SASSYS-1 without any further manipulation. This eliminates the potential errors from calculating material density changes, including errors of sign (i.e. if the perturbation was an *increase* or *decrease* in density).

Ongoing developments in this area include planned support for reactivity feedback coefficients on an arbitrary mesh. Currently, PERSENT must produce results on the same mesh as defined for the SAS4A/SASSYS-1 model. This restriction will be lifted in the future so that each model can use an appropriate mesh for their physics needs.

Improved interaction between PERSENT and SAS4A/SASSYS-1 greatly facilitates the preparation of the SAS4A/SASSYS-1 feedback, improves user efficiency, and reduces the risk of errors. This capability is being utilized for the collaboration with CEA to analyze end-of-cycle transients for the ASTRID sodium cooled fast reactor prototype.

2.3.3 Spatial Kinetics Capabilities

SAS4A/SASSYS-1 can be coupled with DIF3D-K in order to perform system-level transients with spatial kinetics feedback, however this capability is not often used and preparing input is cumbersome. A preliminary input preprocessor has been developed that can read steady-state DIF3D input and generate much of the material composition input needed for SAS4A/SASSYS-1. The preprocessor still requires additional information to map core subassemblies to flow channels and it doesn't resolve all input requirements. Nevertheless, it facilitates the process.

The preprocessor was used to generate input for a coupled model of the EBR-II SHRT-45 benchmark, which is a very detailed and complex model. However, due to limitations of the coupling interface, a somewhat simplified EBR-II model had to be used. Even the simplified model represents a more challenging case than what is typically encountered in conceptual design studies.

During development, a number of issues were discovered and partially resolved. These relate to the more comprehensive data present in the cross section (ISOTXS) and delayed neutron precursor (DLAYXS) files generated by the new MC^2-3 code. MC^2-3 produces ISOTXS files with isotope specific χ vectors rather than a file-wide vector and the coupling interface was not able to handle that case. Also, the DLAYXS file now contains isotope-specific delayed neutron data, but SAS4A/SASSYS-1 can only handle a single file-wide set of six terms. The ISOTXS issue has been resolved, however the DLAYXS issue is not. The developers of MC^2-3 have been notified of the issue with the hope that they will add the capability to generate a single set of delayed neutron precursors per file.

An additional major issue still remains unresolved. Although the EBR-II model appears to successfully complete the steady-state initialization, the problem hangs approximately 30 seconds into the transient. The hang occurs in DIF3D-K itself, which is beyond the scope of this development work. Future effort should be invested to identify and correct the problem.

2.3.4 Steady-State Flow Initialization

SAS4A/SASSYS-1 includes a very powerful steady-state initialization model that simplifies the task of defining a consistent flow network for most reactor designs. The model includes sophisticated search routines that determine pressure and temperature distributions throughout the primary and intermediate systems. From these distributions, pump and heat exchanger requirements can be determined.

In the current ASTRID benchmark specification, an internal reactor vessel cooling flow path is defined that travels from the inlet plenum to an upper annular volume and then back down along the reactor vessel wall. To account for heat transfer between the two opposing flow paths, the SAS4A/SASSYS-1 model developer needed to define a negative flow for one of the parallel paths. This resulted in a dead-end for the search routines and eventual code failure.

The steady-state initialization model has been updated to handle this unusual situation. Effectively, the search routine can now “back out” of a flow path in order to find all routes to complete the flow network and determine pressure drops.

3. Related Work

As described in the introduction, the SAS4A/SASSYS-1 safety analysis code is an important component of a number of programs in the DOE Office of Advanced Reactor Technology. It is also supported by two external users: TerraPower, LLC and the Korea Atomic Energy Research Institute (KAERI). Not only do the external collaborations provide support for additional development, they also provide valuable feedback on code usage and functionality. This section provides a brief summary of the sponsored development activities and illustrates the benefit of the collaborations to DOE.

TerraPower, LLC sponsored considerable developments and documentation efforts on the spatial kinetics coupling capabilities between SAS4A/SASSYS-1 and DIF3D-K. Developments include the capability to model physical control rod motion during a transient. Individual control rods can be manipulated through input to evaluate transient behavior and power distortions of a large core design. In addition, the exchange of external reactivity feedback effects was added to include global effects such as radial core expansion, control-rod driveline expansion, and axial expansion, which are difficult to represent in the fixed lattice used by DIF3D-K. TerraPower also supported preparation of documentation for the coupling interface.

During the development of the control-rod motion model for TerraPower, a need to support generalized input tables was identified. Two types of tables were required for the new model: moveable control rod compositions and control rod position vs. time. Rather than the fixed-size tables traditionally used by SAS4A/SASSYS-1, a new TABLE input block has been created that will dynamically adjust to any size input. Although the new block was developed for the control-rod motion model, it has already been identified as a key component for improving the flexibility of all table input in SAS4A/SASSYS-1. In particular, it can be used to improve the reactivity feedback input (see Section 2.3.2).

TerraPower also supported development of a dual fuel reactivity feedback model in which a single core assembly may contain different fuel enrichments. The intent of the new model is to better characterize significant axial enrichment variations in high burnup fuel. The model is now being used to evaluate the axially heterogeneous ASTRID core design. Previously, SAS4A/SASSYS-1 was unable to represent the feedback effect of an internal axial blanket, so there has been a significant benefit from this collaboration.

These two new capabilities have been incorporated into a beta for version 5.1 and have been provided to TerraPower for testing.

KAERI sponsors considerable development on metallic fuel performance and severe accident modeling. These new models represent a long-term effort and are currently being incorporated into a separate branch of SAS4A/SASSYS-1. Developments presented in this report are being merged into the separate branch to ensure continuous compatibility of the two versions. Once the KAERI developments are mature, they can be readily reintegrated into the main development branch of SAS4A/SASSYS-1. This will make the new models available to all users.

4. Summary

Modernization of the data management in the SAS4A/SASSYS-1 source code and updates to several other code modules were completed in FY14 to ensure that SAS4A/SASSYS-1 remains a viable simulation tool for the safety analysis of advanced, low-pressure, non-LWR reactor concepts. The availability and continued modernization of SAS4A/SASSYS-1 is vital to the on-going missions of several DOE-NE programs and will reduce the steep learning curve for future code developers.

In addition to code modernization, important new extensions have been introduced to address needed capabilities. The most significant of these is the development of a coupling interface that allows SAS4A/SASSYS-1 to interact with external power conversion system models. This capability will be used to couple SAS4A/SASSYS-1 with the CO_2 Brayton cycle Plant Dynamics Code, which will allow analysts the ability to assess full plant responses to a variety of load demands or system upset conditions.

SAS4A/SASSYS-1 has a slowly growing user base that will strengthen the promotion of advanced reactor concepts such as sodium cooled fast reactors. Additional users will also help solidify DOE's leadership role in fast reactor safety both domestically and in international collaborations.

5. References

1. R. Schmidt, et al., *Sodium Fast Reactor Gaps Analysis of Computer Codes and Models for Accident Analysis and Reactor Safety*, SAND2011-4145, Sandia National Laboratories, June 2011.
2. T. Sofu and L. Briggs, "Benchmark Specifications for EBR-II Shutdown Heat Removal Tests," *Proc. ICAPP '12*, Chicago, IL, June 24-28, 2012.
3. L. L. Briggs et al., "EBR-II Passive Safety Demonstration Tests Benchmark Analyses – Phase 1", *Trans. Amer. Nucl. Soc.*, Anaheim, CA, **111**, pp. 1263-1266, November 9-13, 2014.
4. T. Sumner, T. H. Fanning, and T. Sofu, *unpublished information*, Argonne National Laboratory, March 2013.
5. T. Sumner, T.K. Kim, and C. Grandy, *unpublished information*, Argonne National Laboratory, September 2013.
6. D. Grabaskas and T. Sofu, *Review of Existing Approaches for Modeling the Reliability of Passive Systems*, ANL-SMR-5, Argonne National Laboratory, September 30, 2013.
7. A. Hakobyan, T. Aldemir, R. Denning, S. Dunagan, D. Kunsman, B. Rutt, U. Catalyurek, "Dynamic Generation of Accident Progression Event Trees," *Nuclear Engineering and Design*, vol. 238, pp. 3457-3467, 2008.

8. T. H. Fanning, F. E. Dunn, D. Grabaskas, T. Sumner, and J.W. Thomas, "SAS4A/SASSYS-1 Modernization," ANL-ARC-265, Argonne National Laboratory, September 20, 2013.
9. A. Moisseytsev & J. Sienicki, "Dynamic Analysis of S-CO_2 Cycle Control with Coupled PDC-SAS4A/SASSYS-1 Codes," *Proceedings of the 20th International Conference on Nuclear Engineering* (ICONE20), Anaheim, CA, July 30 – August, 2012.

Nuclear Engineering Division
Argonne National Laboratory
9700 South Cass Avenue, Bldg. 208
Argonne, IL 60439-4842

www.anl.gov